

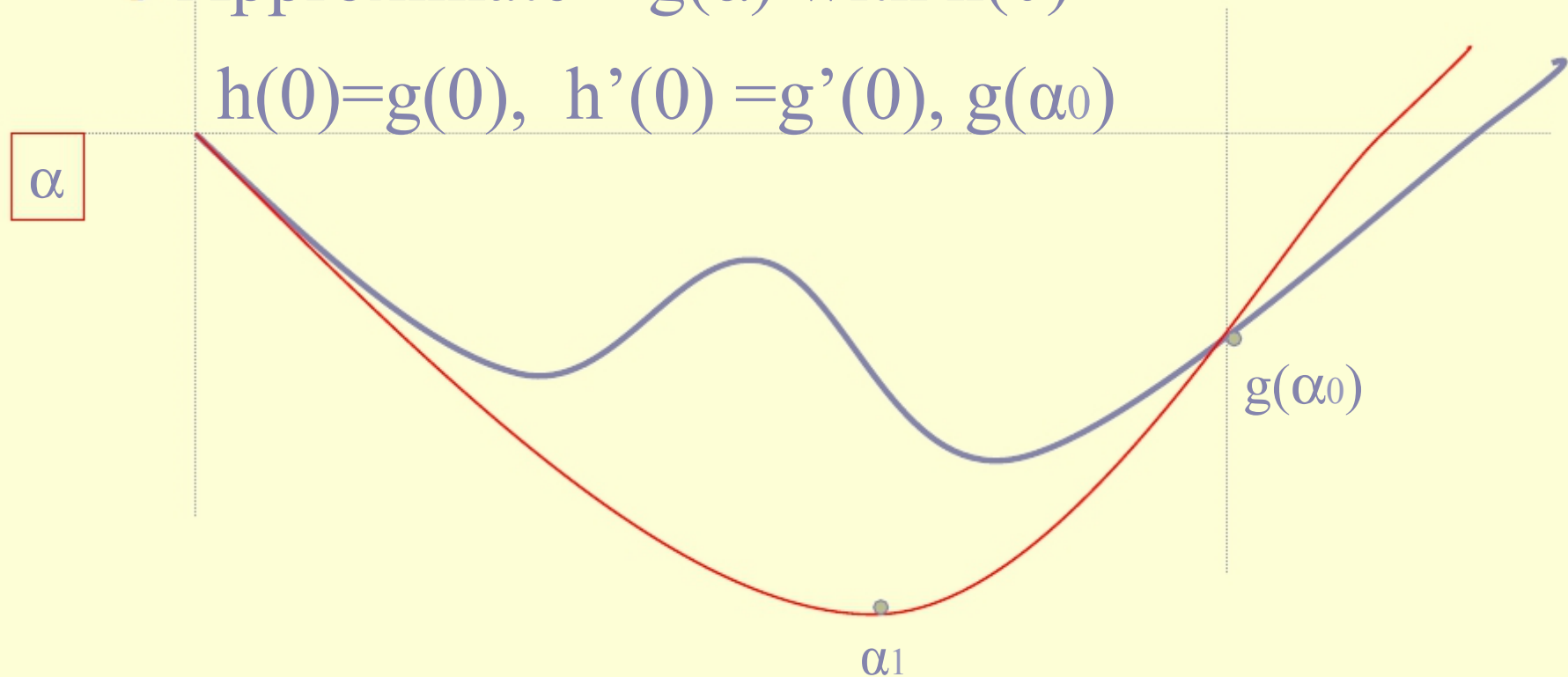
3.2.2 LINE SEARCH METHODS: EXTRAS.

3.2.2.1 LINE SEARCH METHODS: USING INTERPOLATION IN LINE SEARCH

Quadratic Interpolation

Approximate $g(\alpha)$ with $h(\alpha)$

$$h(0)=g(0), \quad h'(0)=g'(0), \quad g(\alpha_0)$$

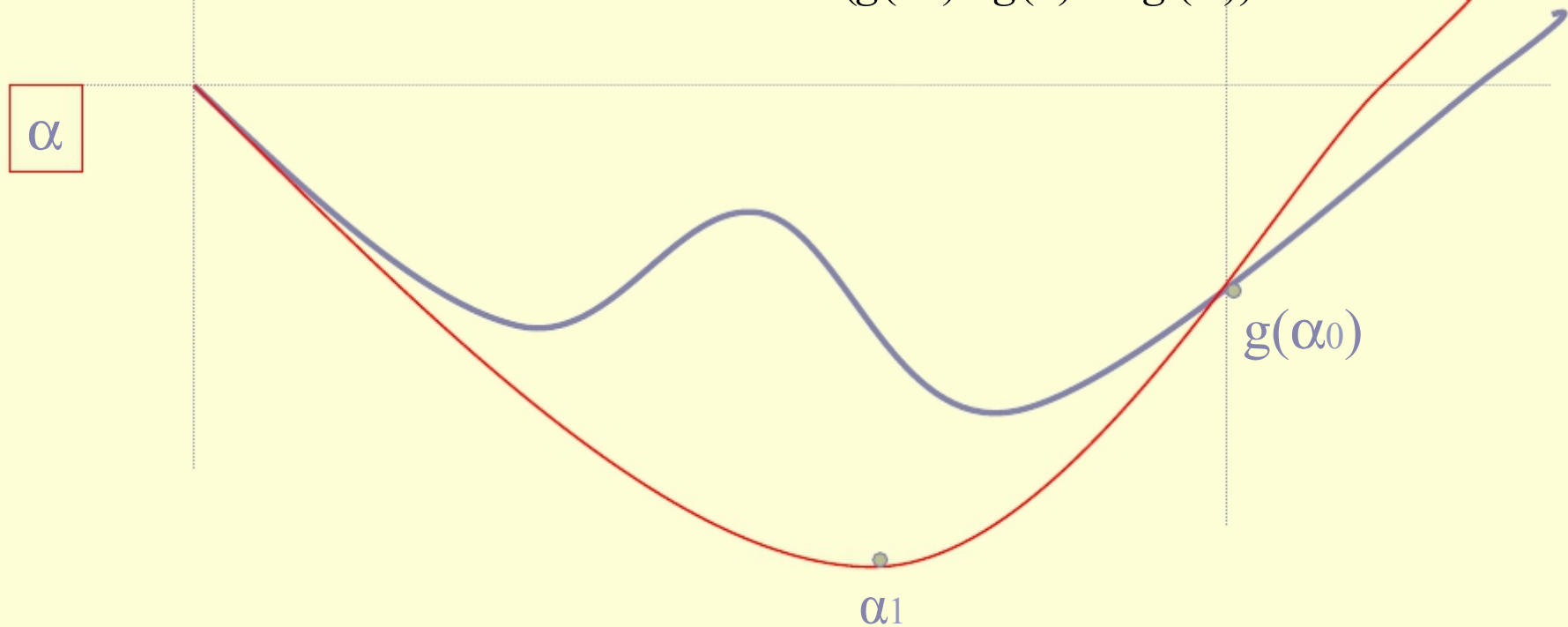


$$\left(\frac{g(\alpha_0) - g(0) - \alpha_0 g'(0)}{2} \right)^2$$

Quadratic Interpolation

Potential step

$$\alpha_1 = \frac{g'(\alpha)\alpha_0^2}{2(g(\alpha_0) - g(0) - \alpha_0 g'(\alpha))}$$



Cubic Interpolation

$$h(\alpha) = a\alpha^3 + b\alpha^2 + \alpha g'(0) + g(0)$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_0^2 \alpha_1^2 (\alpha_1 - \alpha_0)} \begin{bmatrix} \alpha_0^2 & -\alpha_1^2 \\ -\alpha_0^3 & \alpha_1^3 \end{bmatrix} \begin{bmatrix} g(\alpha_1) - g(0) - g'(0)\alpha_1 \\ g(\alpha_0) - g(0) - g'(0)\alpha_0 \end{bmatrix}$$

$$\alpha_2 = \frac{-b \pm \sqrt{b^2 - 3ag'(0)}}{3a}$$

3.2.2.2 LINE SEARCH METHODS: OTHER LINE SEARCH PRINCIPLES

Unconstrained optimization methods

$$x_{k+1} = x_k + \alpha_k d_k$$

α_k : Step length

d_k : Search direction

1) Line search

2) Trust-Region algorithms

Quadratic approximation

Influences

Step length computation:

1) Armijo rule:

$$f(x_k) - f(x_k + \beta^m \tau_k d_k) \geq -\rho \beta^m \tau_k \nabla f(x_k)^T d_k$$

$$\beta \in (0,1) \quad \rho \in (0,1/2)$$

$$\tau_k = -(\nabla f(x_k)^T d_k) / \|d_k\|^2$$

2) Goldstein rule:

$$\rho_1 \alpha_k g_k^T d_k \leq f(x_k + \alpha_k d_k) - f(x_k) \leq \rho_2 \alpha_k g_k^T d_k$$

$$0 < \rho_2 < \frac{1}{2} < \rho_1 < 1$$

3) Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k$$

$$0 < \rho \leq \sigma < 1$$

Implementations:

Shanno (1978)

More - Thiente (1992-1994)

3.2.2.3 LINE SEARCH

METHODS: TAXONOMY OF METHODS

Methods for Unconstrained Optimization

1) Steepest descent (Cauchy, 1847)

$$d_k = -\nabla f(x_k)$$

2) Newton

$$d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

3) Quasi-Newton (Broyden, 1965; and many others)

$$d_k = H_k \nabla f(x_k)$$

4) Conjugate Gradient Methods (1952)

$$d_{k+1} = -g_{k+1} + \beta_k s_k$$

β_k is known as the conjugate gradient parameter

$$s_k = x_{k+1} - x_k$$

5) Truncated Newton method (Dembo, et al, 1982)

$$d_k \cong -\nabla^2 f(x_k)^{-1} g_k \qquad \|r_k\| = \|\nabla^2 f(x_k) d_k + g_k\|$$

6) Trust Region methods

7) Conic model method (Davidon, 1980)

$$q(d) = f(x_k) + g_k^T d + \frac{1}{2} d^T B_k d$$

$$c(d) = f(x_k) + \frac{g_k^T d}{1 + b^T d} + \frac{1}{2} \frac{d^T A_k d}{(1 + b^T d)^2}$$

8) Tensorial methods (Schnabel & Frank, 1984)

$$m_T(x_c + d) = f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2} \nabla^2 f(x_c) \cdot d^2 \\ + \frac{1}{6} T_c \cdot d^3 + \frac{1}{24} V_c \cdot d^4$$

9) Methods based on systems of Differential Equations

Gradient flow Method (Courant, 1942)

$$\frac{dx}{dt} = -\alpha \nabla^2 f(x)^{-1} \nabla f(x)$$

$$x(0) = x_0$$

10) Direct searching methods

Hooke-Jeeves (form searching) (1961)

Powell (conjugate directions) (1964)

Rosenbrock (coordinate system rotation)(1960)

Nelder-Mead (rolling the simplex) (1965)

Powell –UOBYQA (quadratic approximation) (1994-2000)

N. Andrei, *Critica Retiunii Algoritmilor de Optimizare fara Restrictii*
Editura Academiei Romane, 2008.

3.3 DEALING WITH INDEFINITE HESSIANS MATRICES

Closest Positive Definite Matrix

- But Hessian is positive definite (maybe) ONLY at solution!!

What do we do?

- Answer: Perturb the matrix.
- Frobenius NORM
- Closest Positive Definite Matrix (symmetric A)

$$\|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2} = \sqrt{\text{tr}(A^* A)} = \sqrt{\sum_{i=1}^n \sigma_i^2}$$

$$A = A^T \Rightarrow \|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2} = \sqrt{\text{tr}(A^2)} = \sqrt{\sum_{i=1}^n \lambda_i^2}$$

$$Q_1^T Q_1 = Q_2^T Q_2 \Rightarrow \|Q_1 A Q_2\|_F = \|A\|_F$$

$$A = Q D Q^T \longrightarrow A_1 = Q B Q^T$$

$$B = \begin{cases} \lambda_i & \lambda_i \geq \delta > 0 \\ \delta & \lambda_i < \delta \end{cases}$$

Given initial point x_0 ;

for $k = 0, 1, 2, \dots$

Factorize the matrix $B_k = \nabla^2 f(x_k) + E_k$, where $E_k = 0$ if $\nabla^2 f(x_k)$ is sufficiently positive definite; otherwise, E_k is chosen to ensure that B_k is sufficiently positive definite;

Solve $B_k p_k = -\nabla f(x_k)$;

Set $x_{k+1} \leftarrow x_k + \alpha_k p_k$, where α_k satisfies the Wolfe, Goldstein, or Armijo backtracking conditions;

end

1. Adding Multiple of the Identity

Algorithm 3.3 (Cholesky with Added Multiple of the Identity).

Choose $\beta > 0$;

if $\min_i a_{ii} > 0$

 set $\tau_0 \leftarrow 0$;

else

$\tau_0 = -\min(a_{ii}) + \beta$;

end (if)

for $k = 0, 1, 2, \dots$

 Attempt to apply the Cholesky algorithm to obtain $LL^T = A + \tau_k I$;

if the factorization is completed successfully

stop and return L ;

else

$\tau_{k+1} \leftarrow \max(2\tau_k, \beta)$;

end (if)

end (for)

- Q: what may be the downside of the approach?

2. Modified Cholesky

```

for    $j = 1, 2, \dots, n$ 
     $c_{jj} \leftarrow a_{jj} - \sum_{s=1}^{j-1} d_s l_{js}^2;$ 
     $d_j \leftarrow c_{jj};$ 
    for    $i = j + 1, \dots, n$ 
         $c_{ij} \leftarrow a_{ij} - \sum_{s=1}^{j-1} d_s l_{is} l_{js};$ 
         $l_{ij} \leftarrow c_{ij} / d_j;$ 
    end
end

```

• Ensuring Quality of the Modified Factorization
(i.e. entries do not blow up by division to small elements)

• AIM:

$$d_j \geq \delta, \quad |m_{ij}| \leq \beta, \quad i = j + 1, j + 2, \dots, n,$$

• Solution: Once a “too small d” is encountered
Replace its value by :

$$d_j = \max \left(|c_{jj}|, \left(\frac{\theta_j}{\beta} \right)^2, \delta \right), \quad \text{with } \theta_j = \max_{j < i \leq n} |c_{ij}|$$

• Then:

$$|m_{ij}| = |l_{ij} \sqrt{d_j}| = \frac{|c_{ij}|}{\sqrt{d_j}} \leq \frac{|c_{ij}| \beta}{\theta_j} \leq \beta, \quad \text{for all } i > j.$$

• Q: Cholesky does not need pivoting. But
does it make sense here to NOT pivot?

$$B_k d = -\nabla f(x_k) \Leftrightarrow LDL^T d = \nabla f(x_k)$$

$$B_k = \nabla_{xx}^2 f(x_k) + E_k$$

LDL factorization WITH permutation (why?)

- EXPAND

3. Modified LDLT (maybe most practical to implement ?)

- What seems to be a practical perturbation to PD that makes it have smallest eigenvalue Delta?
- Solution: Keep same L,P, modify only the B!

$$PAP^T = LBL^T$$

I will ask you to
code it with
Armijo

$$F = Q \operatorname{diag}(\tau_i) Q^T, \quad \tau_i = \begin{cases} 0, & \lambda_i \geq \delta, \\ \delta - \lambda_i, & \lambda_i < \delta, \end{cases} \quad i = 1, 2, \dots, n,$$

$$P(A + E)P^T = L(B + F)L^T, \quad \text{where } E = P^T L F L^T P.$$

3.4 QUASI-NEWTON METHODS

3.4 QUASI-NEWTON METHODS: ESSENTIALS

Secant Method – Derivation (NLE)

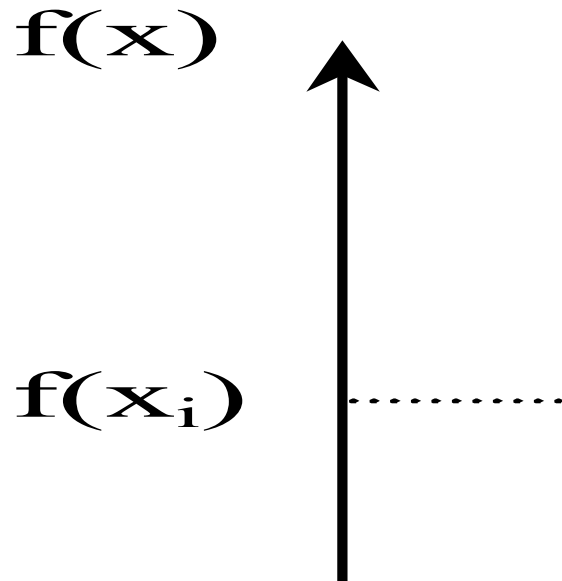


Figure 1 Geometrical illustration of the Newton-Raphson method.

$$f(x) = 0$$

Newton's Method

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (1)$$

Approximate the derivative

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \quad (2)$$

Substituting Equation (2) into Equation (1) gives the Secant method

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

Secant Method – Derivation

The secant method can also be derived from geometry:

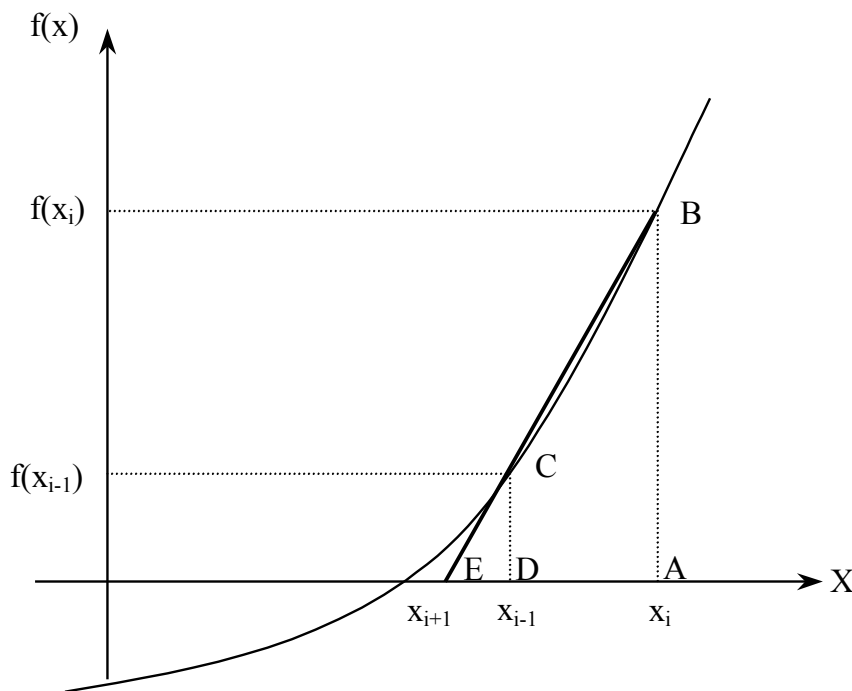


Figure 2 Geometrical representation of the Secant method.

The Geometric Similar Triangles

$$\frac{AB}{AE} = \frac{DC}{DE}$$

can be written as

$$\frac{f(x_i)}{x_i - x_{i+1}} = \frac{f(x_{i-1})}{x_{i-1} - x_{i+1}}$$

On rearranging, the secant method is given as

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

Multidimensional Secant Conditions

Given two points x_k and x_{k+1} , we define (for an optimization problem)

$$g_k = \nabla f(x_k) \quad \text{and} \quad g_{k+1} = \nabla f(x_{k+1})$$

Further, let $p_k = x_{k+1} - x_k$, then

$$g_{k+1} - g_k \approx H(x_k) p_k \quad \leftarrow \text{The Secant Condition}$$

If the Hessian is constant, then

$$g_{k+1} - g_k = H p_k \quad \text{which can be rewritten as} \quad q_k = H p_k$$

If the Hessian is constant, then the following condition would hold as well

$$H^{-1}_{k+1} q_i = p_i \quad 0 \leq i \leq k$$

This is called the quasi-Newton condition.

Broyden–Fletcher–Goldfarb–Shanno

Remember that $\mathbf{q}_i = \mathbf{H}_{k+1} \mathbf{p}_i$ and $\mathbf{H}_{k+1}^{-1} \mathbf{q}_i = \mathbf{p}_i$ (or, $\mathbf{B}_{k+1} \mathbf{q}_i = \mathbf{p}_i$) $0 \leq i \leq k$

Both equations have exactly the same form, except that \mathbf{q}_i and \mathbf{p}_i are interchanged and \mathbf{H} is replaced by \mathbf{B} (or vice versa).

This leads to the observation that any update formula for \mathbf{B} can be transformed into a corresponding complimentary formula for \mathbf{H} by interchanging the roles of \mathbf{B} and \mathbf{H} and of \mathbf{q} and \mathbf{p} . The reverse is also true.

Broyden–Fletcher–Goldfarb–Shanno formula update of \mathbf{H}_k is obtained by taking the complimentary formula of the DFP formula, thus:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{q}_k \mathbf{q}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} - \frac{\mathbf{H}_k \mathbf{p}_k \mathbf{p}_k^T \mathbf{H}_k}{\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k}$$

By taking the inverse, the BFGS update formula for \mathbf{B}_{k+1} (i.e., \mathbf{H}_{k+1}^{-1}) is obtained:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \left(\frac{1 + \mathbf{q}_k^T \mathbf{B}_k \mathbf{q}_k}{\mathbf{q}_k^T \mathbf{p}_k} \right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} - \frac{\mathbf{p}_k \mathbf{q}_k^T \mathbf{B}_k + \mathbf{B}_k \mathbf{q}_k \mathbf{p}_k^T}{\mathbf{q}_k^T \mathbf{p}_k}$$

How do BFGS methods work?

LEMMA 8.1.1 Let $A \in \mathbb{R}^{n \times n}$, $s, y \in \mathbb{R}^n$, $s \neq 0$. Then for any matrix norms $\|\cdot\|$, $\|\cdot\|$ such that

$$\|A \cdot B\| \leq \|A\| \cdot \|B\| \quad (8.1.6)$$

and

$$\left\| \frac{vv^T}{v^T v} \right\| = 1, \quad (8.1.7)$$

the solution to

$$\min_{B \in Q(y, s)} \|B - A\| \quad (8.1.8)$$

is

$$A_+ = A + \frac{(y - As)s^T}{s^T s}. \quad (8.1.9)$$

In particular, (8.1.9) solves (8.1.8) when $\|\cdot\|$ is the l_2 matrix norm, and (8.1.9) solves (8.1.8) uniquely when $\|\cdot\|$ is the Frobenius norm.

Advantage of quasi-Newton

- Matrix is ALWAYS positive definite, so line search works fine.
- It needs ONLY gradient information.
- It behaves *almost* like Newton in the limit (convergence is superlinear).
- Optimality is nice, but what you really need is that (1) it needs only derivatives and (2) it satisfies the secant property and (3) if the original matrix is PSD so is the update. I will ask you to prove them.
- In its L-BFGS variant it is the workhorse of weather forecast and operational data assimilation in general (a max likelihood procedure, really).

4.1 TRUST REGION FUNDAMENTALS

Trust Region Idea

- Notations

$$f^k = f(x^k) \quad \nabla f^k = \nabla f(x^k)$$

- Quadratic Model

- Order of Quadratic Model (Taylor)
- $$m_k(p) = f^k + p^T g^k + \frac{1}{2} p^T B^k p$$

$$f(x^k + p) = f^k + p^T g^k + \frac{1}{2} p^T \nabla_{xx}^2 f(x^k + tp) p \quad t \in [0,1]$$

$$m_k(p) - f(x^k + p) = \begin{cases} O(\|p\|^2) \\ O(\|p\|^2) & B^k = \nabla_{xx}^2 f(x^k) \end{cases}$$

Trust Region Subproblem

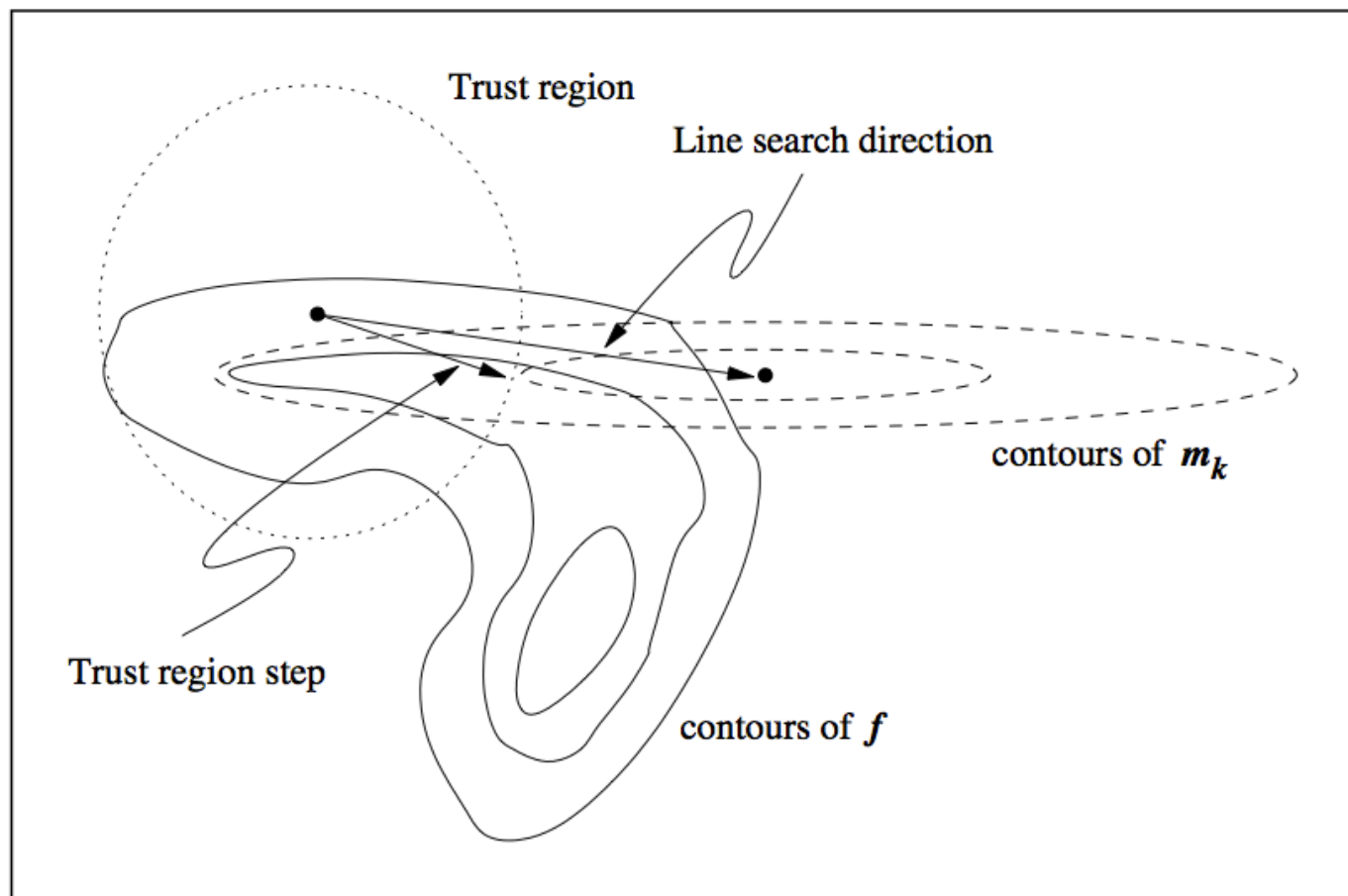
$$\begin{array}{ll} \min_{p \in \mathbb{R}^n} & m_k(p) \\ \text{subject to} & \|p\| \leq \Delta^k \end{array}$$



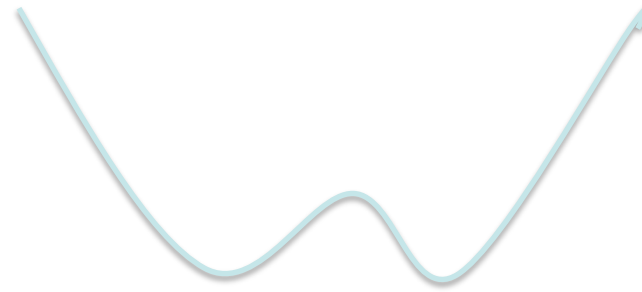
Called Trust Region
Constraint

- If $B^k \succ 0$ and $p^{*k} = (B^k)^{-1} g^k$; where $\|p^{*k}\| \leq \Delta^k$ then p^k is the solution of the TR subproblem.
- But the interesting case lies in the opposite situation (since not, why would you need the TR in first place)?

Trust Region Geometric Intuition



$$\min_x (x^2 - 1)^2$$



- Line search started at 0 cannot progress.
- How about the trust-region?

$$\min_d -2d^2; \quad |d| \leq \Delta$$

- Either solution will escape the saddle point -- that is the principle of trust-region.

- How do we solve the TR subproblem?
- If $B^k \succ 0$ (or if we are not obsessed with stopping at saddle points) we use “dogleg” method. (LS, NLE). Most linear algebra is in computing

$$B^k d^{k,U} = -g^k$$

- If fear saddle points, we have to mess around with eigenvalues and eigenvectors – much harder problem.

Trust Region Management: Parameters

- The quality of the reduction.

$$\rho^k = \frac{f(x^k) - f(x^k + p^k)}{m_k(0) - m_k(p^k)}$$

Actual Reduction

Predicted Reduction

- Define the maximum TR size $\eta \in \left[0, \frac{1}{4}\right)$

$$\hat{\Delta}; \quad \Delta \in [0, \hat{\Delta})$$

Algorithm 4.1 (TRUST REGION).

Given $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, and $\eta \in [0, \frac{1}{4})$:

for $k = 0, 1, 2, \dots$

Obtain p_k by (approximately) solving (4.3);

Evaluate ρ_k from (4.4);

if $\rho_k < \frac{1}{4}$

$$\Delta_{k+1} = \frac{1}{4} \Delta_k$$

else

if $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$

$$\Delta_{k+1} = \min(2\Delta_k, \hat{\Delta})$$

else

$$\Delta_{k+1} = \Delta_k;$$

if $\rho_k > \eta$

$$x_{k+1} = x_k + p_k$$

else

$$x_{k+1} = x_k;$$

I will ask you to
code it with
dogleg

What if I cannot solve the TR exactly ?

- Since it is a hard problem.
- Will this destroy the “Global” convergence behavior?
- Idea: Accept a “sufficient” reduction.
- But, I have no Armijo (or Wolfe, Goldshtein criterion) ...
- What do I do?
- Idea? Solve a simple TR problem that creates the yardstick for acceptance – the Cauchy point.

4.2 THE CAUCHY POINT

The Cauchy Point

- What is an easy model to solve? Linear model

$$l_k(p) = f^k + g^{k,T} p$$

- Solve TR linear model

$$p^{k,s} = \arg \min_{p \in R^n} _{\|p\| \leq \Delta^k} l_k(p)$$

- The Cauchy point.

$$\tau^k = \arg \min_{\tau \in R} _{\|\tau p^{k,s}\| \leq \Delta^k} m_k(\tau p^{k,s})$$

$$p^{k,c} = \tau^k p^{k,s}; \quad x^{k,c} = x^k + p^{k,c}$$

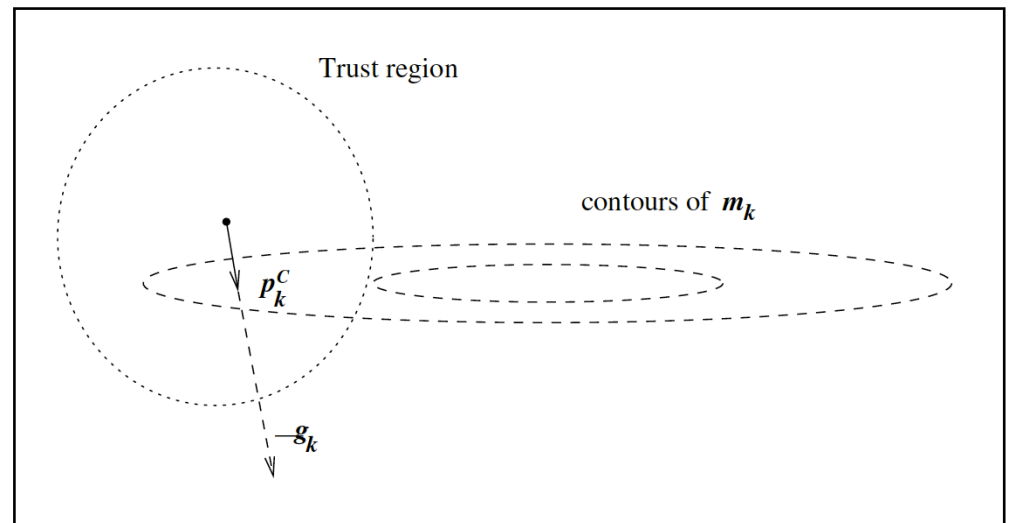
- The reduction $m(0) - m(p^{k,c})$ becomes my yardstick; if trust region has at least this decrease, I can guarantee “global” convergence (reduction is $O(\|g^k\|^2)$)

Cauchy Point Solution

- First, solution of the linear problem is

$$p_k^s = -\frac{\Delta^k}{\|g^k\|} g^k$$

- Then, it immediately follows that



$$\tau_k = \begin{cases} 1 & g_k^T B_k g_k \leq 0 \\ \min\left(\frac{\|g_k\|^3}{(g_k^T B_k g_k) \Delta_k}, 1\right) & \text{otherwise} \end{cases}$$

Dogleg Methods: Improve CP

- If Cauchy point is on the boundary I have a lot of decrease and I accept it (e.g if $g^{k,T} B_k g^k \leq 0$;))
- If Cauchy point is interior,

$$g^{k,T} B_k g^k > 0; \quad p^{k,c} = -\frac{\|g_k\|^2}{g^{k,T} B_k g^k} g^k$$

- Take now “Newton” step $p^B = -B_k^{-1} g^k$ (note, B need not be pd, all I need is nonsingular).

Dogleg Method Continued

I will ask you to
code it with TR

- Define dogleg path

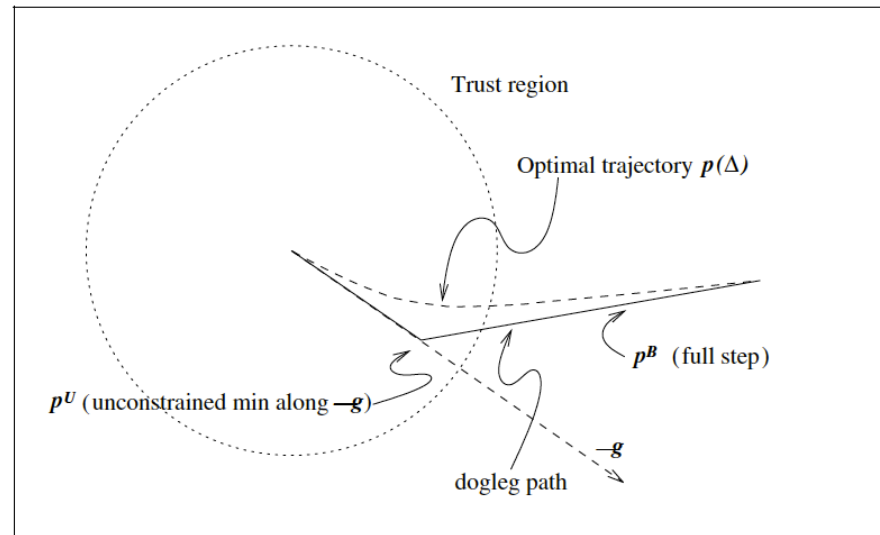
$$\tilde{p}(\tau) = \begin{cases} \tau p^{k,c} & \tau \leq 1 \\ p^{k,c} + (\tau - 1)(p^B - p^{k,c}) & 1 \leq \tau \leq 2 \end{cases}$$

- The dogleg point:

$$\tilde{p}(\tau_D); \quad \tau_D = \arg \min_{\tau; \|\tilde{p}(\tau)\| \leq \Delta_k} m_k(\tilde{p}(\tau))$$

- It is obtained by solving 2 quadratics.
- Sufficiently close to the solution it allows me to choose the Newton step, $\tau = 2$ and thus quadratic convergence.

Dogleg Method: Theory



Lemma 4.2.

Let B be positive definite. Then

- (i) $\|\tilde{p}(\tau)\|$ is an increasing function of τ , and
- (ii) $m(\tilde{p}(\tau))$ is a decreasing function of τ .

Global Convergence of CP Methods

Lemma 4.3.

The Cauchy point p_k^c satisfies (4.20) with $c_1 = \frac{1}{2}$, that is,

$$m_k(0) - m_k(p_k^c) \geq \frac{1}{2} \|g_k\| \min \left(\Delta_k, \frac{\|g_k\|}{\|B_k\|} \right).$$

$$\|p_k\| \leq \gamma \Delta_k, \quad \text{for some constant } \gamma \geq 1. \quad (4.25)$$

$$m_k(0) - m_k(p_k) \geq c_1 \|g_k\| \min \left(\Delta_k, \frac{\|g_k\|}{\|B_k\|} \right), \quad (4.20)$$

Theorem 4.5.

Let $\eta = 0$ in Algorithm 4.1. Suppose that $\|B_k\| \leq \beta$ for some constant β , that f is bounded below on the level set S defined by (4.24) and Lipschitz continuously differentiable in the neighborhood $S(R_0)$ for some $R_0 > 0$, and that all approximate solutions of (4.3) satisfy the inequalities (4.20) and (4.25), for some positive constants c_1 and γ . We then have

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (4.26)$$

Numerical comparison between methods

- What is a fair comparison between methods?
- Probably : starting from same point 1) number of function evaluations and 2) number of linear systems (the rest depends too much on the hardware and software platform). I will ask you to do this.
- Trust region tends to use fewer function evaluations (the modern preferred metric;) than line search .
- Also dogleg does not force positive definite matrix, so it has fewer chances of stopping at a saddle point, (but it is not guaranteed either).

4.3 GENERAL CASE: SOLVING THE ACTUAL TR PROBLEM (DOGLEG DOES NOT QUITE DO IT)

Trust Region Equation

Theorem 4.1.

The vector p^ is a global solution of the trust-region problem*

$$\min_{p \in \mathbb{R}^n} m(p) = f + g^T p + \frac{1}{2} p^T B p, \quad \text{s.t. } \|p\| \leq \Delta, \quad (4.7)$$

if and only if p^ is feasible and there is a scalar $\lambda \geq 0$ such that the following conditions are satisfied:*

$$(B + \lambda I)p^* = -g, \quad (4.8a)$$

$$\lambda(\Delta - \|p^*\|) = 0, \quad (4.8b)$$

$$(B + \lambda I) \quad \text{is positive semidefinite.} \quad (4.8c)$$

Theory of Trust Region Problem

Global convergence
away from saddle
point

Theorem 4.8.

Suppose that the assumptions of Theorem 4.6 are satisfied and in addition that f is twice continuously differentiable in the level set S . Suppose that $B_k = \nabla^2 f(x_k)$ for all k , and that the approximate solution p_k of (4.3) at each iteration satisfies (4.52) for some fixed $\gamma > 0$. Then $\lim_{k \rightarrow \infty} \|g_k\| = 0$.

If, in addition, the level set S of (4.24) is compact, then either the algorithm terminates at a point x_k at which the second-order necessary conditions (Theorem 2.3) for a local solution hold, or else $\{x_k\}$ has a limit point x^ in S at which the second-order necessary conditions hold.*

Fast Local
Convergence

Theorem 4.9.

Let f be twice Lipschitz continuously differentiable in a neighborhood of a point x^ at which second-order sufficient conditions (Theorem 2.4) are satisfied. Suppose the sequence $\{x_k\}$ converges to x^* and that for all k sufficiently large, the trust-region algorithm based on (4.3) with $B_k = \nabla^2 f(x_k)$ chooses steps p_k that satisfy the Cauchy-point-based model reduction criterion (4.20) and are asymptotically similar to Newton steps p_k^N whenever $\|p_k^N\| \leq \frac{1}{2}\Delta_k$, that is,*

$$\|p_k - p_k^N\| = o(\|p_k^N\|). \quad (4.53)$$

Then the trust-region bound Δ_k becomes inactive for all k sufficiently large and the sequence $\{x_k\}$ converges superlinearly to x^ .*

How do we solve the subproblem?

- Very sophisticated approach based on theorem on structure of TR solution, eigenvalue analysis and/or an “inner” Newton iteration.
- Foundation: Find Solution for

$$p(\lambda) = -(B + \lambda I)^{-1}g$$

$$\|p(\lambda)\| = \Delta.$$

How do I find such a solution?

$$B = Q\Lambda Q^T \quad \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n),$$

$$p(\lambda) = -Q(\Lambda + \lambda I)^{-1}Q^T g = -\sum_{j=1}^n \frac{q_j^T g}{\lambda_j + \lambda} q_j,$$

, by orthonormality of q_1, q_2, \dots, q_n :

$$\|p(\lambda)\|^2 = \sum_{j=1}^n \frac{(q_j^T g)^2}{(\lambda_j + \lambda)^2}.$$

TR problem has a solution

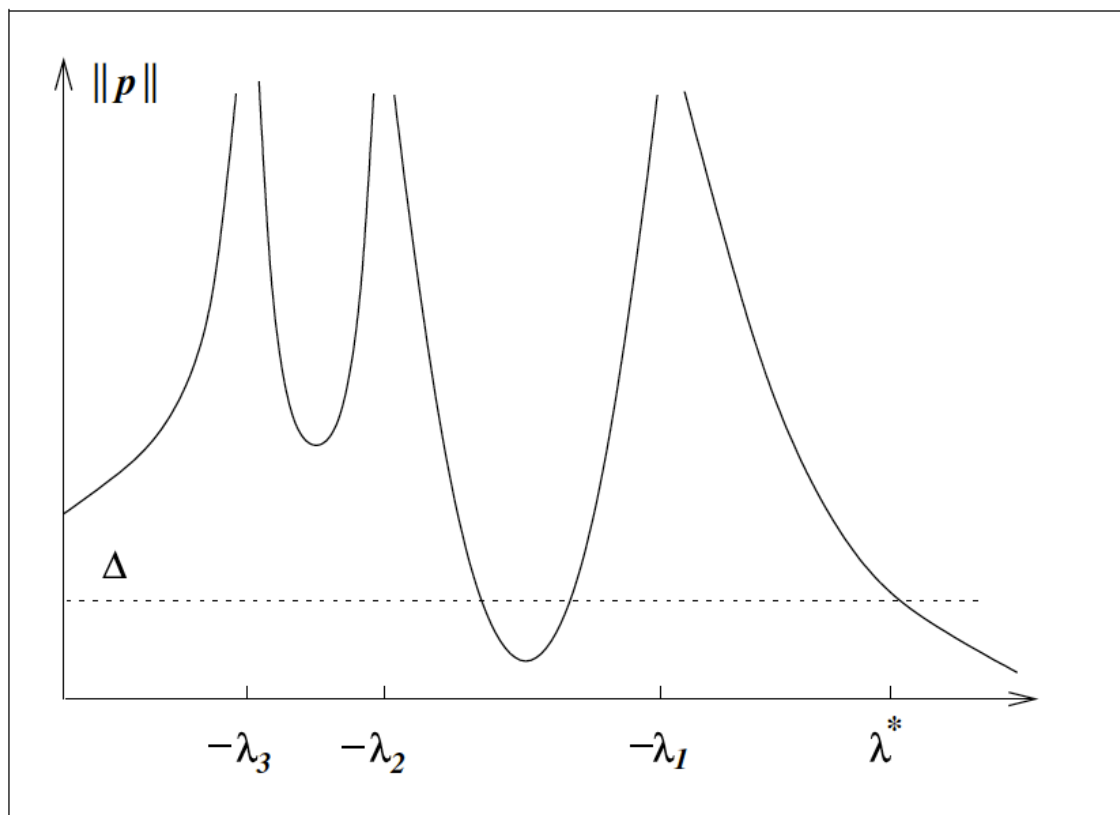


Figure 4.5 $\|p(\lambda)\|$ as a function of λ .

$$\lim_{\lambda \rightarrow \infty} \|p(\lambda)\| = 0. \quad q_j^T g \neq 0 \Rightarrow \lim_{\lambda \rightarrow -\lambda_j} \|p(\lambda)\| = \infty.$$

Practical (INCOMPLETE) algorithm

$$\phi_2(\lambda) = \frac{1}{\Delta} - \frac{1}{\|p(\lambda)\|}, \quad \lambda^{(\ell+1)} = \lambda^{(\ell)} - \frac{\phi_2(\lambda^{(\ell)})}{\phi_2'(\lambda^{(\ell)})}.$$

Algorithm 4.3 (Trust Region Subproblem).

Given $\lambda^{(0)}$, $\Delta > 0$:

for $\ell = 0, 1, 2, \dots$

Factor $B + \lambda^{(\ell)}I = R^T R$;

Solve $R^T R p_\ell = -g$, $R^T q_\ell = p_\ell$;

Set

$$\lambda^{(\ell+1)} = \lambda^{(\ell)} + \left(\frac{\|p_\ell\|}{\|q_\ell\|} \right)^2 \left(\frac{\|p_\ell\| - \Delta}{\Delta} \right);$$

end (for).

It generally gives a machine precision solution in 2-3 iterations
(Cholesky)

The Hard Case

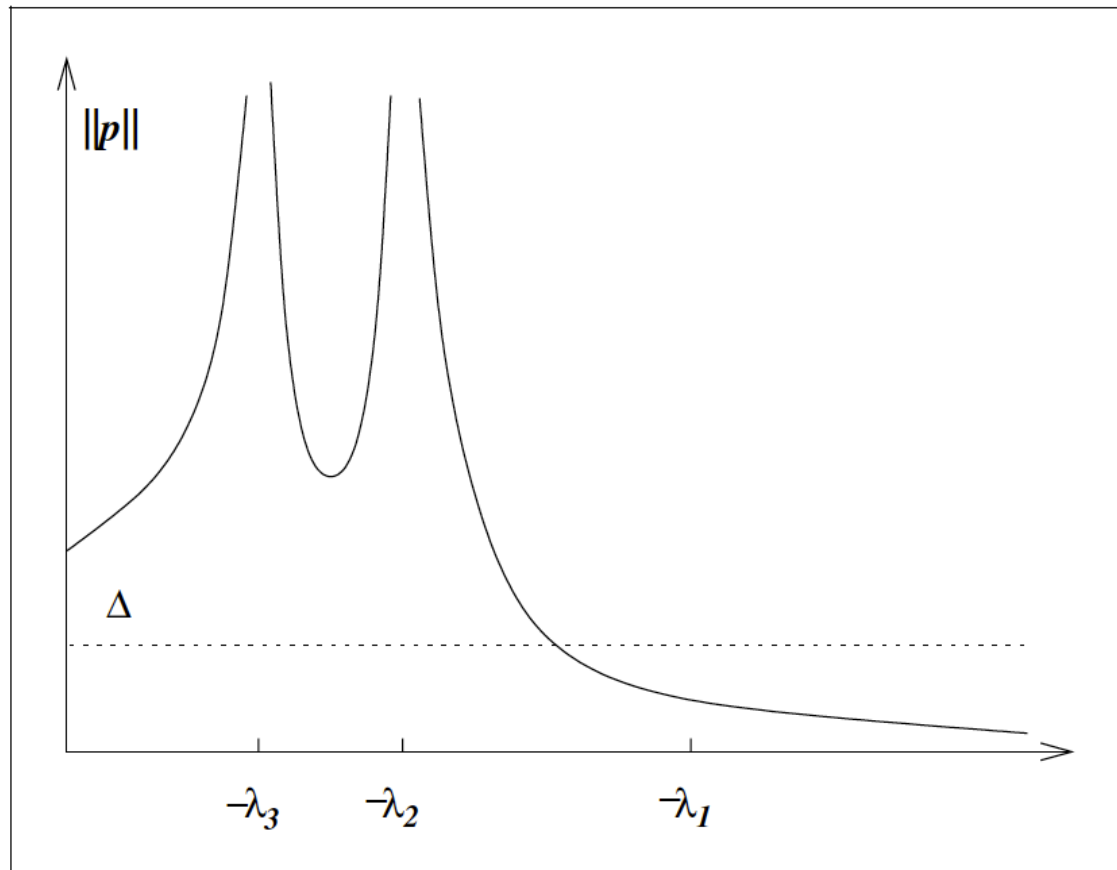


Figure 4.7 The hard case: $\|p(\lambda)\| < \Delta$ for all $\lambda \in (-\lambda_1, \infty)$.

If double root, things continue to be complicated ...

$$q_j^T g = 0$$

$$\lambda = -\lambda_1 \Rightarrow p = \sum_{j:\lambda_j \neq \lambda_1} \frac{q_j^T g}{\lambda_j - \lambda_1} q_j$$



$$p(\tau) = \sum_{j:\lambda_j \neq \lambda_1} \frac{q_j^T g}{\lambda_j - \lambda_1} q_j + \tau q_1$$



$$\exists \tau \quad \|p(\tau)\| = \Delta^k$$

Summary and Comparisons

- Line search problems have easier subproblems (if we modify Cholesky).
- But they cannot be guaranteed to converge to a point with positive semidefinite Hessian.
- Trust-region problems can, at the cost of solving a complicated subproblem.
- Dogleg methods leave “between” these two situations.